官改包教程修订版

一、刷机包结构简析:

一般官方 miui 刷机固件，包括以下文件/夹：
firmware-update:所谓的底层，基带，驱动，传感器之类都在这里，玩包一般不需要改动
META-INF:里面是刷机脚本以及对应的二进制，加上 miui 官方的签名
compatibility. zip 和 boot.img:这个就不多说了
system.new.dat.br、system.patch.dat、system.transfer.list:这 3 个是一对的，作用是把定义好的 system 文件夹刷入到刷机的对应分区里面
vendor.new.dat.br、vendor.patch.dat、vendor.transfer.list，这是把 vendor 文件夹释放到对应分区
注意:有些包是没有.br 文件的，请灵活运用

二、 官改包改了什么

1、 /system/framework/services.jar:包括核心签名破解(部分)、高级重启(需 /system/media/theme/default/powermenu)、支持 xp 刷入、破解卡米，黑域，秒截图等
2、 /system/app/ThemeManager/ThemeManager. apk 主题商店破解
/system/app/miuisystem/miuisystem.apk 防止主题恢复默认
3、 /system/priv-app/Settings/Settings.apk 精简设置内用户手册，小米商城、加入高级 设置推广
4、 /system/priv-app/MiuiSystemUI/MiuiSystemUI. apk 时间居中，显秒，状态栏天气，系统级圆角
5、 /system/framework/conscrypt.jar 与/system/framework/core-oj.jar 签名破解
6、 /system/etc/hosts 修改可去掉部分官方与启动页广告
7、 加 id，验证 qq 号，将推广写入内核之类

三、 高级设置运行原理

其类型大致分为两种：
其一，独立的程序控制（移动叔叔，bbk,极光之类），再有就是深度集成进设置界面内（例如俄罗斯版官方包，酷安某大佬的定制包）。
前者一般是将指令写入到独立的 apk 内，之后执行 shell（或脚本）命令，进而实现文件的移动，重命名，复制，修改操作。该方式便于推广，适合大量出包而不要单独适配对应的高级设置（通常需要超级权限）
后者是将官方系统进行局部编译修改，注入对应 smail，从而实现系统内切换，且无需超级权限，但需单独适配机型
举个例子:黑域的实现
众人皆知，黑域本质就是修改了 system/framework/services. jar 这个文件。既然这样，我 们只需要提前给改文件打好补丁，封装在某文件夹内，再执行 shell 命令，将事先处理好的 文件复制进去，就实现了开启黑域，同理，关闭也是一样

四、 打包简析

就是将我们修改好的文件，打包成可以让第三方 rec 识别并写入到我们手机的系 列过程
具体来说：system 文件夹-img 文件-dat 文件-dat.br 文件（非绝对），vendor 分区同理
说下解包，本质就是把如上流程反一下

五、 updater-script 简析

该文件位于:META-INF/com/google/android 目录下。此脚本为文本文件，可用文本编辑软 件直接编辑，打开该文件，不难发现该脚本大致分为了 3 部分:机型验证，一般位于第一行，其余的为基带底层的刷入、system/vendor 分区的刷入。面具的刷入，supersu 的刷入，以及杜比的刷入都需要在此加入对应的脚本。这里需要注意，rec 在刷写镜像时，无法自动跳过错误，因而脚本必须百分百正确。否则会出现各类错误提示（不排除是 update-binary 问题）

二进制脚本:META-INF/com/google/android/update-binary

以下展示具体实例:

```
#机型验证
getprop("ro.product.device") == "dipper" || abort("E3004: This package is for \"dipper\" devices; this is a \"" + getprop("ro.product.device") + "\".");

#刷机时显示的信息
ui_print("Target: Xiaomi/dipper/dipper:9/PKQ1.180729.001/9.2.21:user/release-keys");

#控制刷机进度条与显示
show_progress(0.100000, 5);
show_progress(0.200000, 10);

#把boot.img刷入到/dev/block/bootdevice/by-name/boot内
package_extract_file("boot.img", "/dev/block/bootdevice/by-name/boot");

#释放system image 。 abort意为停止，释放失败时显示并终止刷机
show_progress(0.600000, 250);
ui_print("Patching system image unconditionally...");
block_image_update("/dev/block/bootdevice/by-name/system", package_extract_file("system.transfer.list"), "system.new.dat.br", "system.patch.dat") ||
  abort("E1001: Failed to update system image.");
ui_print("Patching vendor image unconditionally...");
block_image_update("/dev/block/bootdevice/by-name/vendor", package_extract_file("vendor.transfer.list"), "vendor.new.dat.br", "vendor.patch.dat") ||
  abort("E2001: Failed to update vendor image.");

#刷入底层,具体见以上boot.img的注解
package_extract_file("firmware-update/cmnlib64.img", "/dev/block/bootdevice/by-name/cmnlib64_a");

#利用busybox挂载system与cust分区
run_program("/sbin/busybox", "mount", "/system");
run_program("/sbin/busybox", "mount", "/cust");

#把刷机包内的supersu文件夹释放到/tmp/supersu目录下
package_extract_dir("supersu", "/tmp/supersu");

#利用busybox将"/tmp/supersu/supersu.zip文件解压到/tmp/supersu目录下
run_program("/sbin/busybox", "unzip", "/tmp/supersu/supersu.zip", "META-INF/com/google/android/*", "-d", "/tmp/supersu");

#利用busybox刷入/tmp/supersu/supersu.zip补丁
run_program("/sbin/busybox", "sh", "/tmp/supersu/META-INF/com/google/android/update-binary", "dummy", "1", "/tmp/supersu/supersu.zip");

#删除/cust/app文件夹
delete_recursive("/cust/app");

#卸载cust分区
ifelse(is_mounted("/cust"), unmount("/cust"));
```

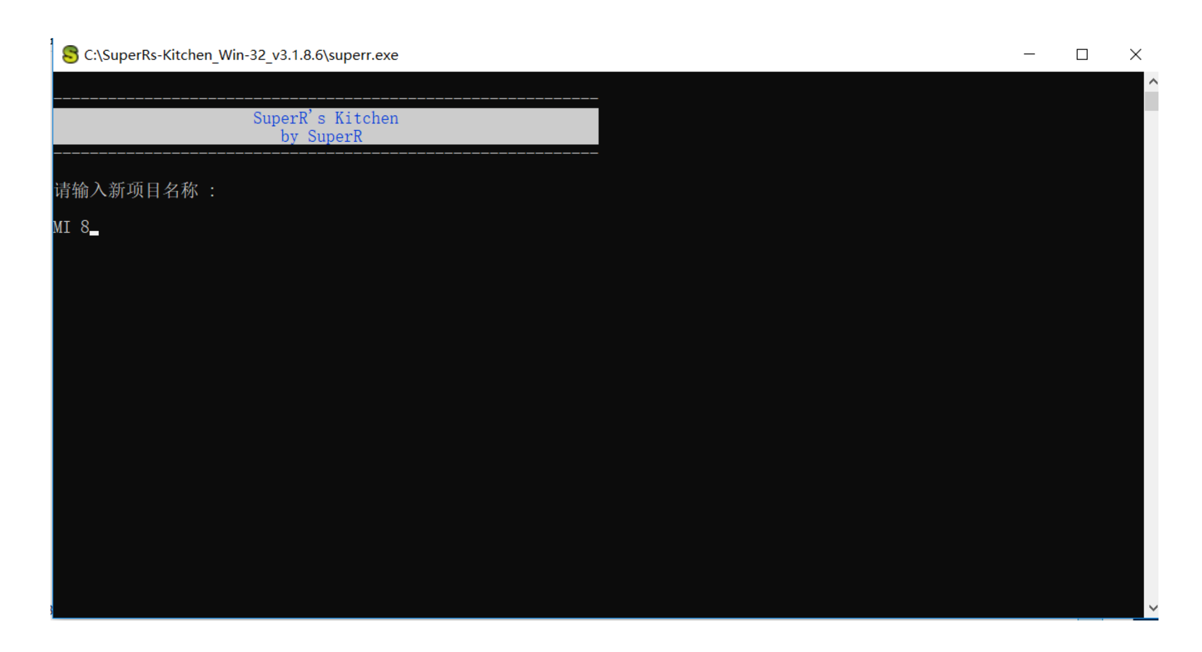
六、简易开机

解包-合并-去卡米-去分区加密-去 dm 校对-打包回去刷机（别忘记去除卡米）

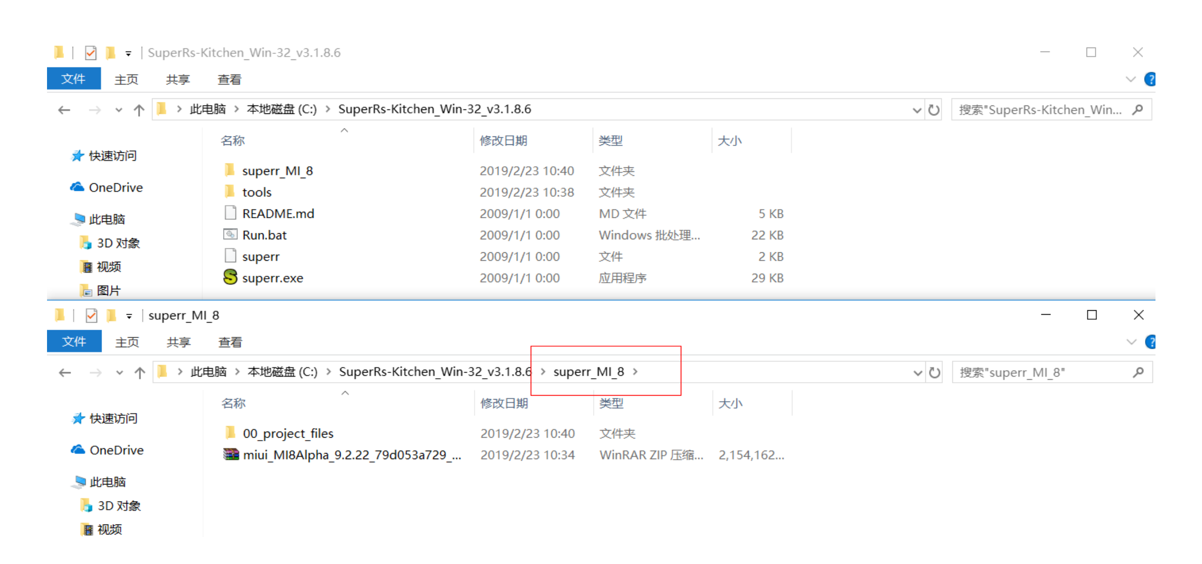
七、厨房实战教程开始

提示：假若你没捐赠版厨房，请看文档结尾解决

新建一个项目，名字随意



复制你的包到工程内



按 4 解包，凡是出现提示均按 y

```
C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe
------------------------------------------------------------
SuperR's Kitchen
by SuperR
------------------------------------------------------------

当前项目:MI_8
安卓版本:放置你的ROM到目录里
然后你按下4就可以解压以及后续操作了

主菜单

1) 创建新项目
2) 选择不同项目
3) 删除一个项目
4) 解压新ROM
5) 检查更新 (当前: v3.1.8.6)
6) Misc工具
7) Boot/Recovery工具
8) ROM工具菜单
9) 插件管理
q = 退出

请选择:
```

这里一般按 3

```
C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe
------------------------------------------------------------
SuperR's Kitchen
by SuperR
------------------------------------------------------------

你想使用哪种刷机脚本授权类型?

1) set_metadata (For KitKat+)
2) set_perm
3) Sparse dat
4) raw_img

请选择:
```

回到主界面，按 8 里面的 1 合并系统

```
C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe
---------------------------------------------------------------
                     SuperR's Kitchen
                       by SuperR
---------------------------------------------------------------

当前项目:MI_8
安卓版本:9
API等级: 28
ARCH: arm64
ARCH2: arm
方法:   vdexExtractor.exe

您要继续deodexing?  y/n
```

按 Y，合并成功

```
C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe
---------------------------------------------------------------
                     SuperR's Kitchen
                       by SuperR
---------------------------------------------------------------

是否删除下列的framework文件夹?  y/n
arm64/arm
```

按 7 里面的 1，解 boot.img

```
C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe

SuperR's Kitchen
by SuperR

当前项目:MI_8
安卓版本:9

主菜单

1) 创建新项目
2) 选择不同项目
3) 删除一个项目
4) 解压新ROM
5) 检查更新 (当前: v3.1.8.6)
6) Misc工具
7) Boot/Recovery工具
8) ROM工具菜单
9) 插件管理
q = 退出

请选择:
```

确保 3、4 选项为绿色显示，之后最小化厨房

```
C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe

SuperR's Kitchen
by SuperR

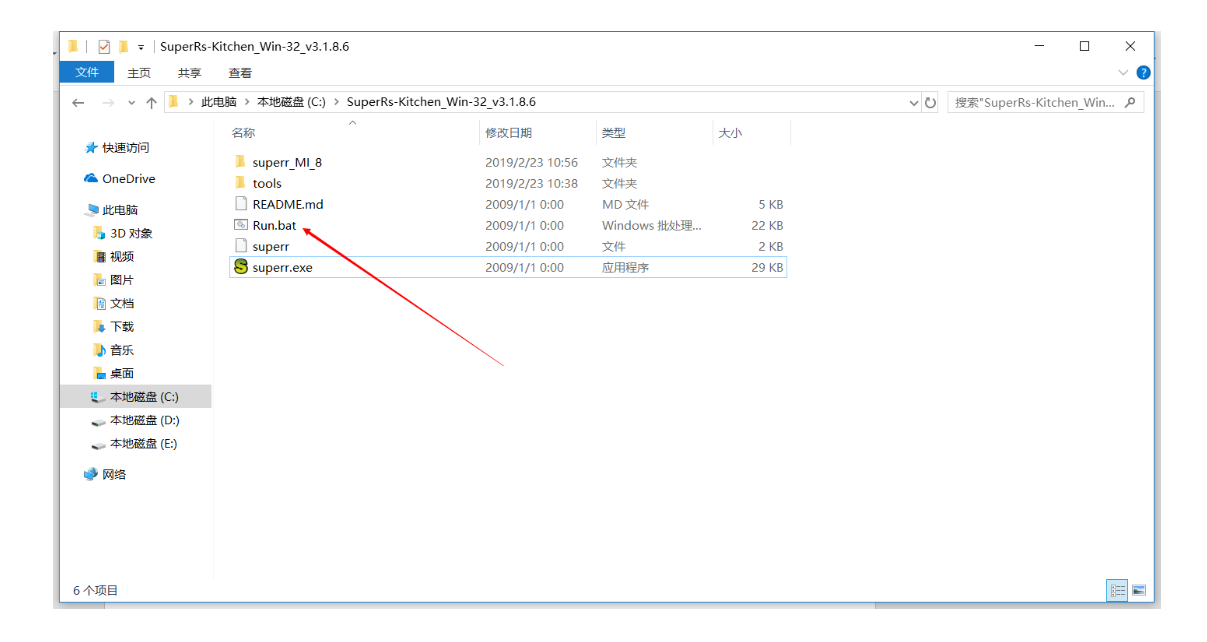
当前项目:MI_8
安卓版本:9

Boot/Recovery工具

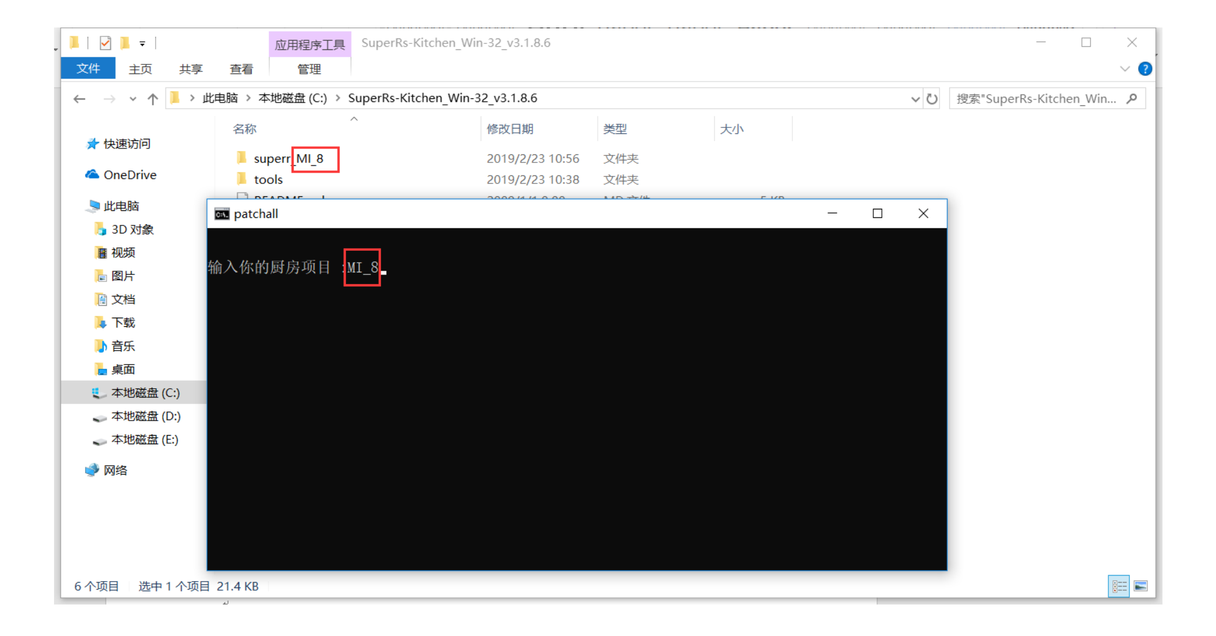
1) 打包boot.img
2) Insecure/Secure 这个boot.img (当前: Secure)
3) 删除 dm-verity (当前: Disabled)
4) 添加/删除 forceencrypt (当前: Disabled)
5) 给sepolicy打补丁--安卓Oreo deodex操作
m = 主菜单
q = 退出

请选择:
```

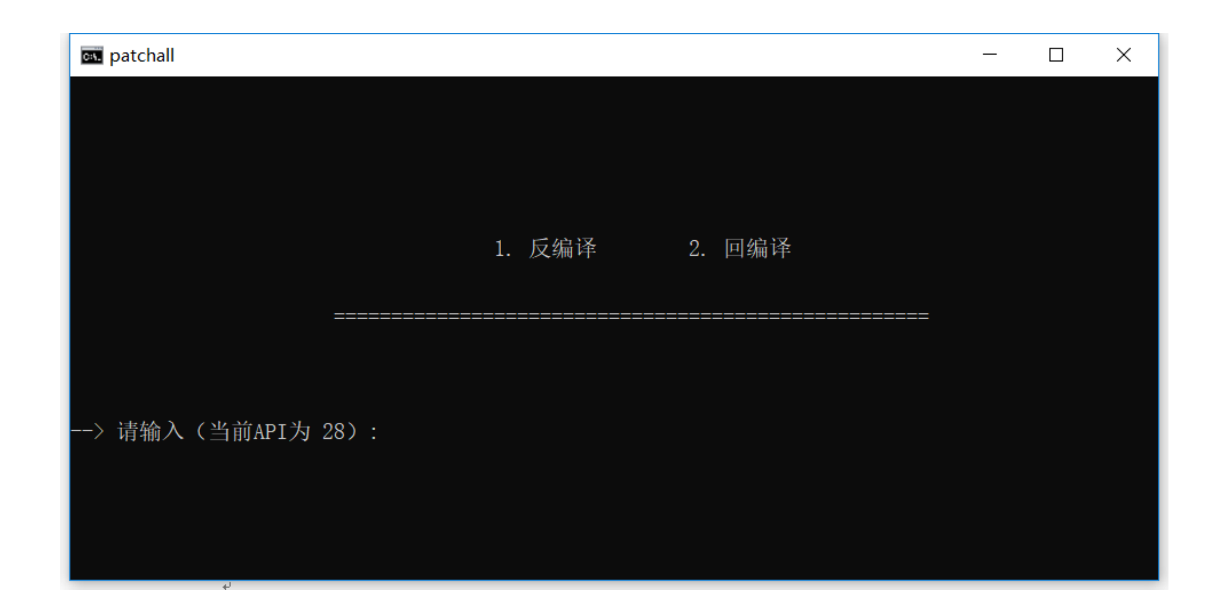
双击 run,bat 启动辅助



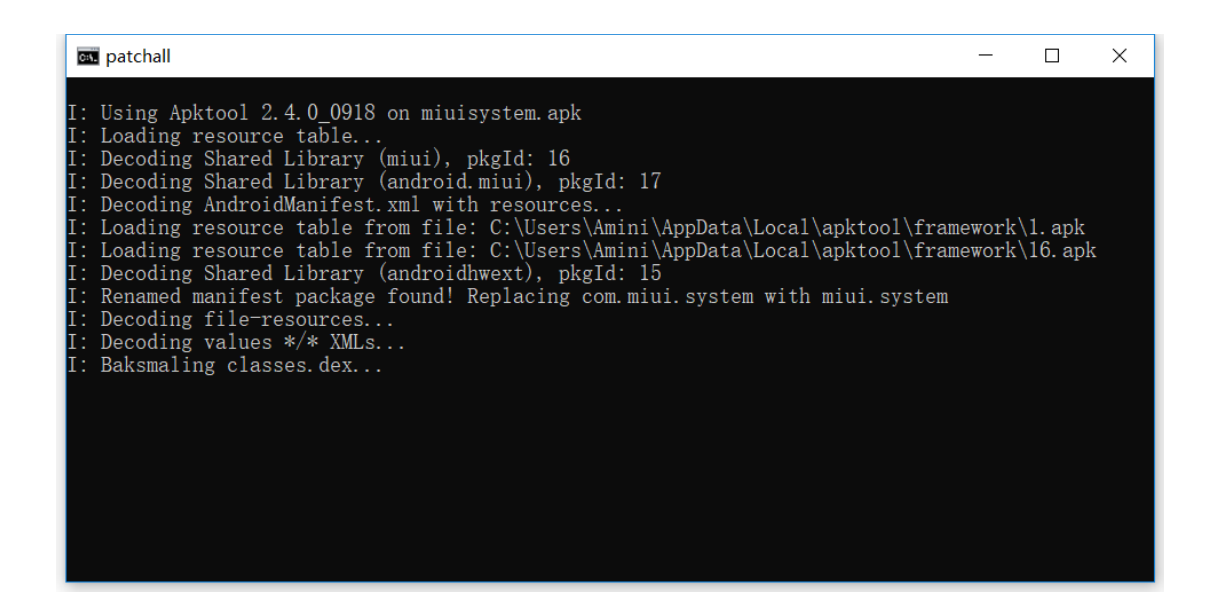
输入厨房项目。自动精简后启动一键辅助



按 1 反编译



自动化操作截图

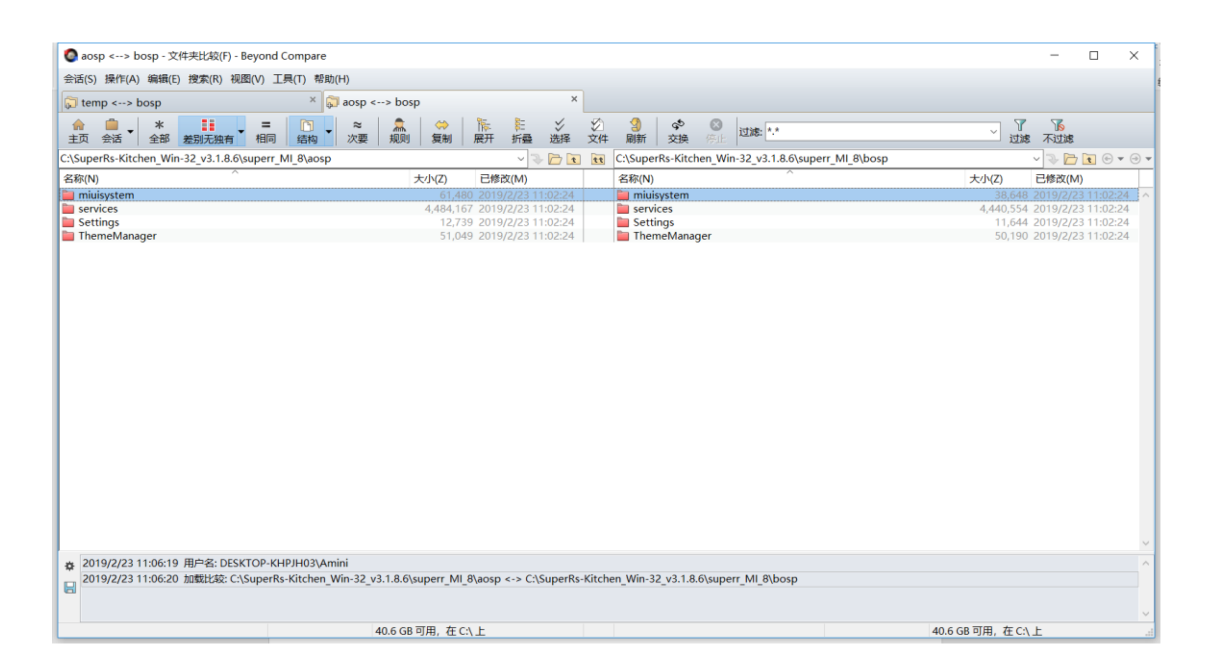


出现以下提示，按任意键自动打开对比器，自行修改 smali/xml 文件（具体修改见文末补充，此处只说破解卡米）

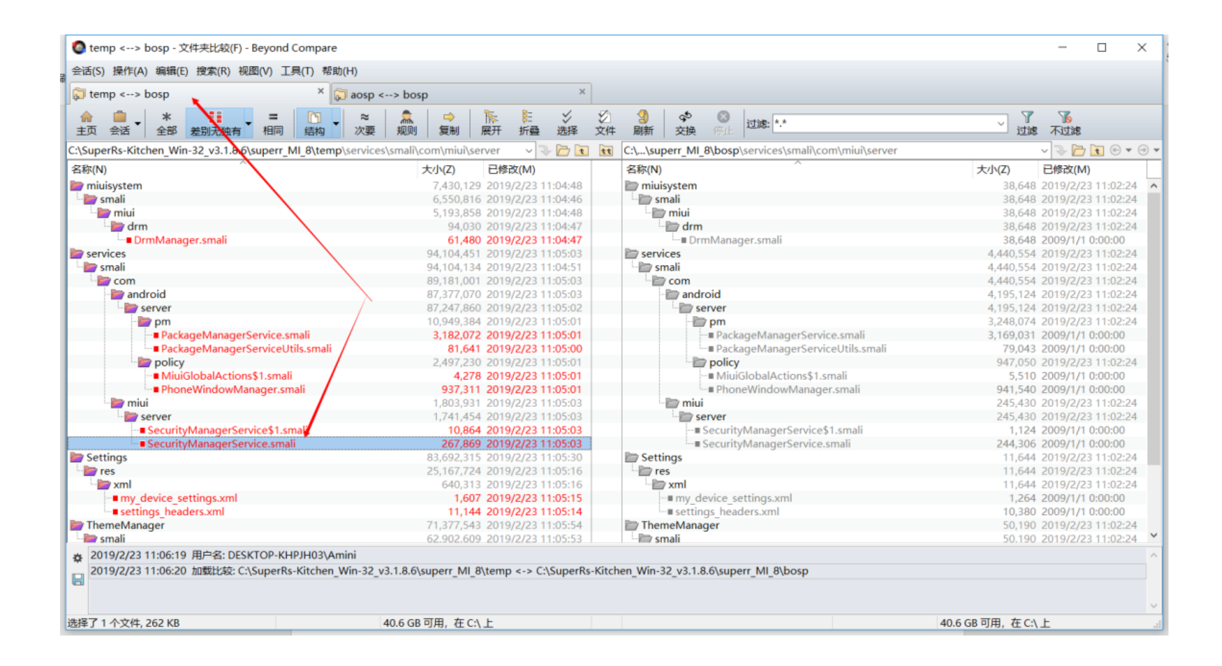
patchall

反编译完成，按任意键继续修改代码. 在回编译完成前，请勿关闭本窗口.

对比器截图

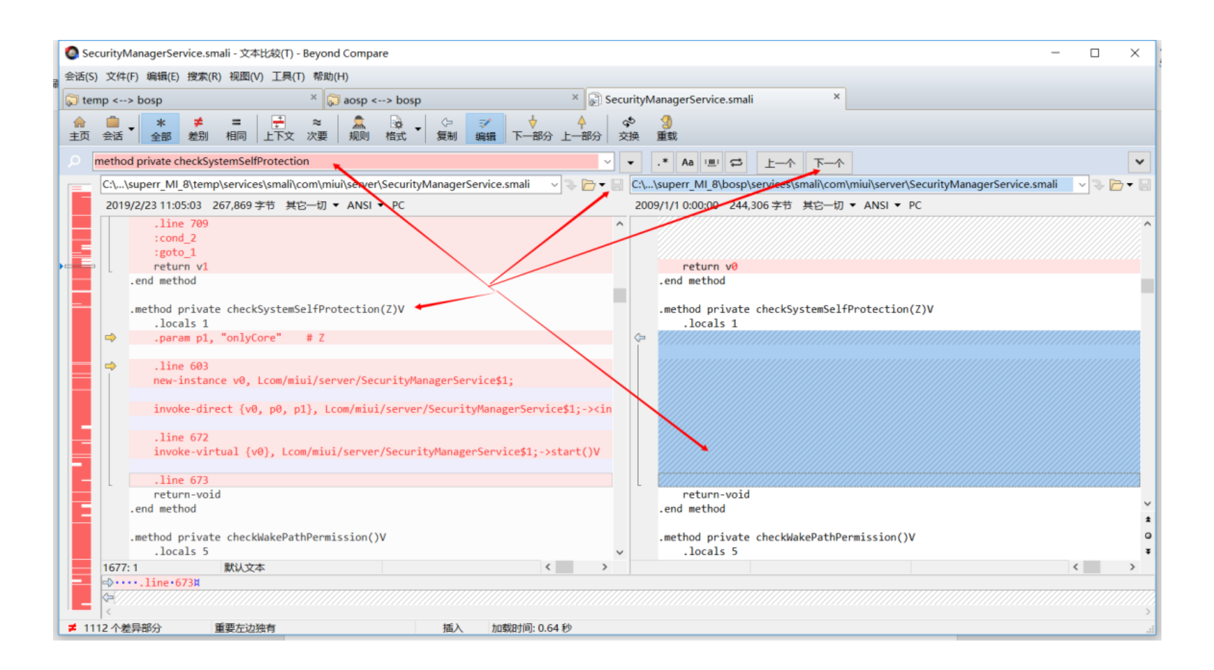


双击该文件

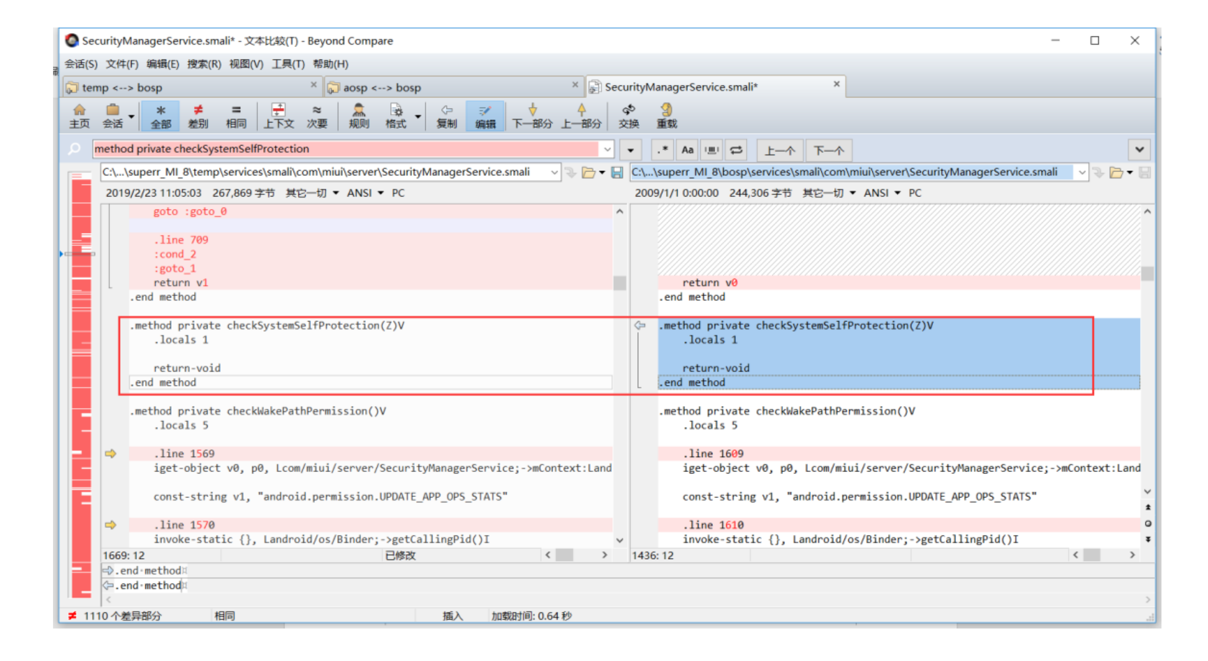


按 ctrl+f，搜索并输入：method private checkSystemSelfProtection，点击下一处

选择右边选中部分，点击小箭头



效果图如下，之后按 ctrl+s 保持修改好，关闭对比器



回到辅助界面，按 2 回编译

patchall

```
I: Using Apktool 2.4.0_0918
I: Smaling smali folder into classes.dex...
I: Building resources...
W: W/ResourceType( 2512): DynamicRefTable(0x10): No mapping for build-time package ID 0x10.
W: W/ResourceType( 2512): e[0x01] -> 0x01
W: W/ResourceType( 2512): e[0x7f] -> 0x7f
W: E/ResourceType( 2512): Failed resolving ResTable_map name at 4136 with ident 0x100b0016
W: warning: Package not found for resource #10010066
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...

I: Using Apktool 2.4.0_0918
I: Smaling smali folder into classes.dex...
```

按任意键，关闭辅助

```
patchall
回编译完成，可以打包啦...
```

打开最小化后的厨房。按 1 打包 boot.img

```
C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe
------------------------------------------------------------
                SuperR's Kitchen
                  by SuperR
------------------------------------------------------------

当前项目:MI_8
安卓版本:9

Boot/Recovery工具

1) 打包boot.img
2) Insecure/Secure 这个boot.img (当前: Secure)
3) 删除 dm-verity (当前: Disabled)
4) 添加/删除 forceencrypt (当前: Disabled)
5) 给sepolicy打补丁--安卓Oreo deodex操作
m = 主菜单
q = 退出

请选择:
```

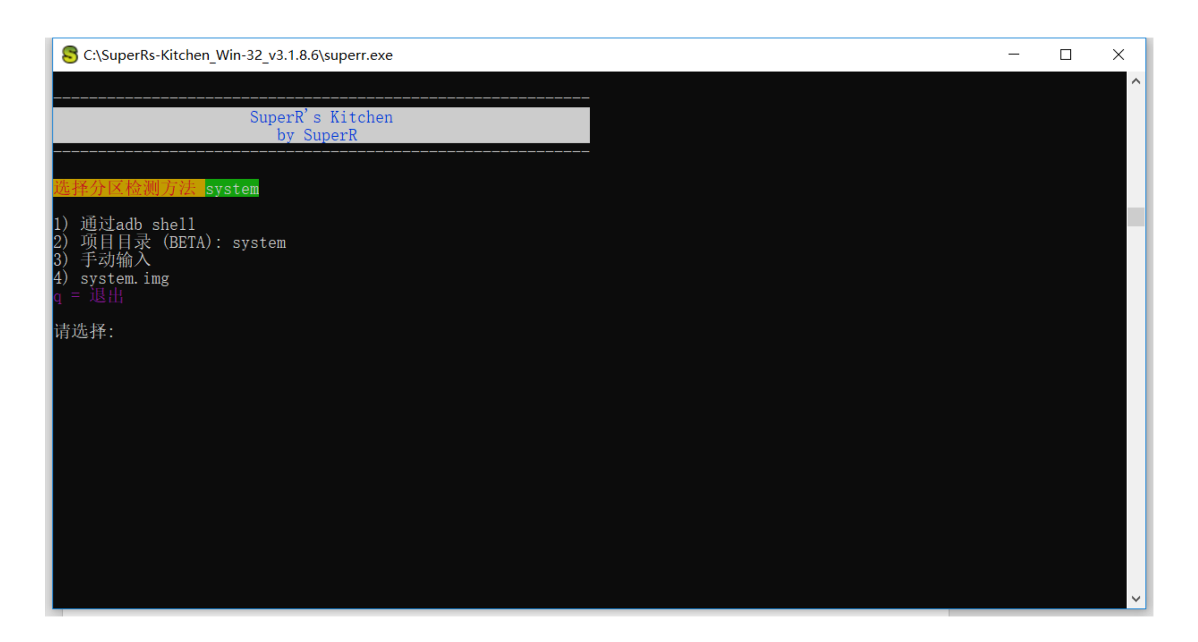
主界面依次按 8、7、1 选项，出现以下提示按 n

C:\SuperRs-Kitchen_Win-32_v3.1.8.6\superr.exe

SuperR's Kitchen
by SuperR

打包ROM之前需要zipalign吗? y/n

按 4 获取 system 分区大小，之后的 vendor 同理



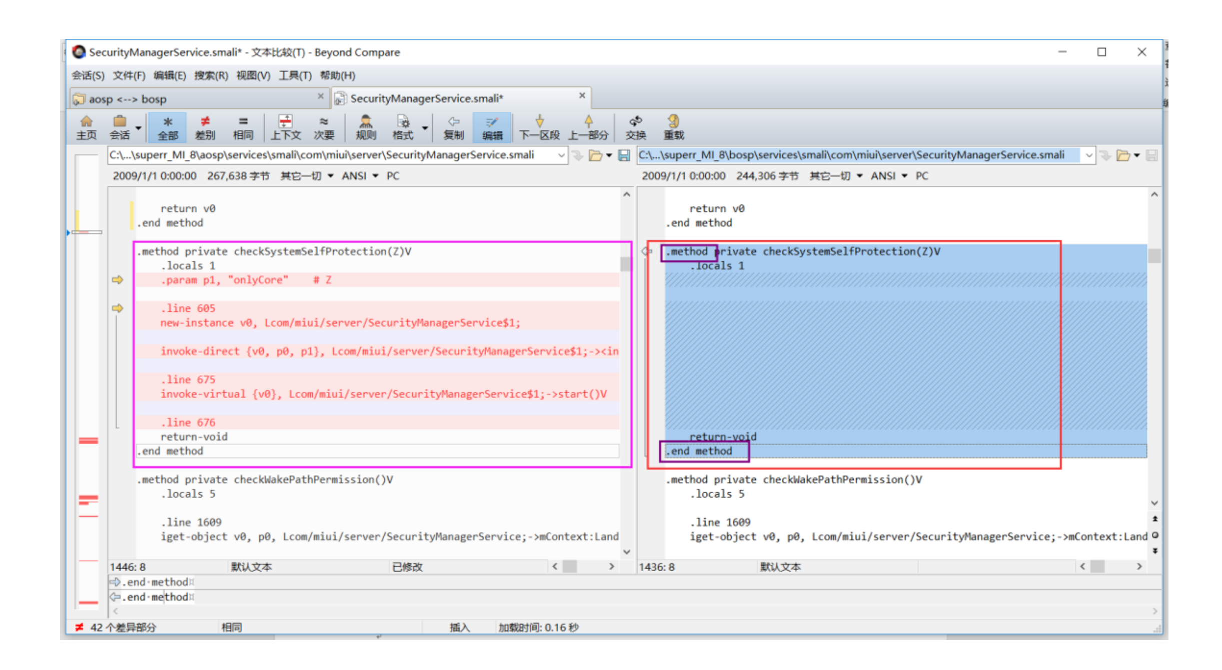
等待自动打包完成，询问签名时，选择 n

教程结束

八、补充教程开始

说下对比器怎么改文件，以上教程中，我们搜索位于 services\smali\com\miui\server\SecurityManagerService.smali 中的一处代码：method private checkSystemSelfProtection （翻译为：检查系统保护。本质就是假若系统文件被删除，则卡米）

那这处代码怎么来的呢？我们在对比器的 aosp 与 boap 对比窗口中，同样打开 SecurityManagerService.smali 文件，如图：



发现差异部分（红色部分）不多，我们点下一部分，就定位到了我们的 method private checkSystemSelfProtection（上面搜索的代码），通过对比 aosp 与 boap 中的 SecurityManagerService.smali 内的差异部分，不难发现，此处无非就是删除了命令本体，所以我们在前面修改时，也用同样的方法修改即可

说明：一个 smali 文件存在一个或多个方法

开始于.method

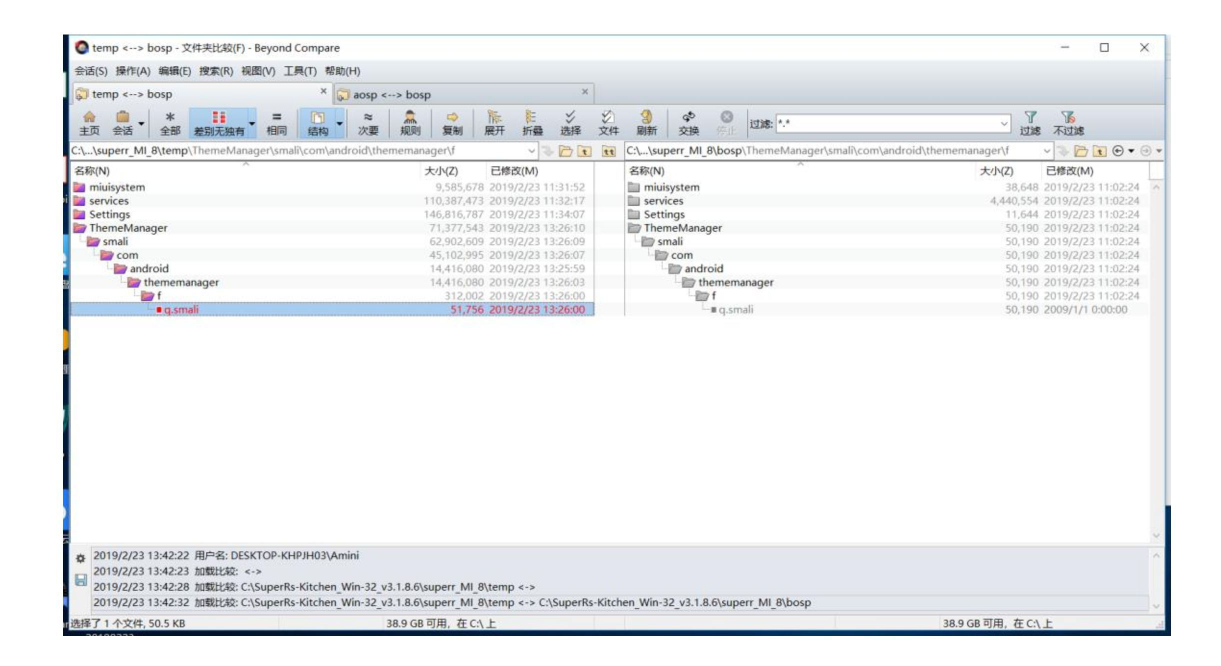
…………………
………………

结束于.end method
（如上图红色方框部分）

对于 miui 官改来说，多半是修改方法中间部分，使得某个值固定为真，而不是判断，或直接返回空值从而达到效果。一般修改，都是以方法（.method 开始）为模块，逐个击破（smali 文件的修改，也可认为是某大佬口中的 patchrom，即插桩,不过是手动的方式）
插桩：通过一种技术将第三方定制的系统功能代码反编译成 smali 注入到将要适配的机器官方 rom 中（cm aosp miui 等）

以下举实例说明：miui 10 最新的主题商店（system\app\ThemeManager\ ThemeManager.apk）的破解流程

打开同一个文件。q.smali



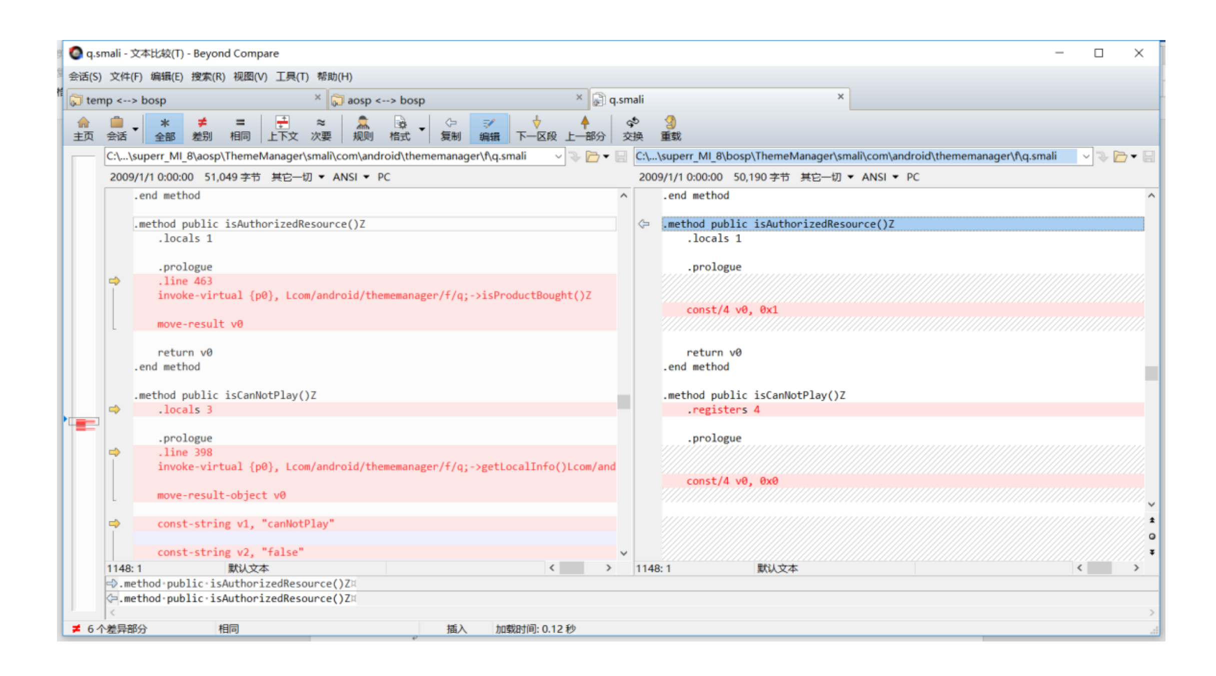
对比 aosp 与 boap 发现，主题破解本质是将

.method public isAuthorizedResource()Z

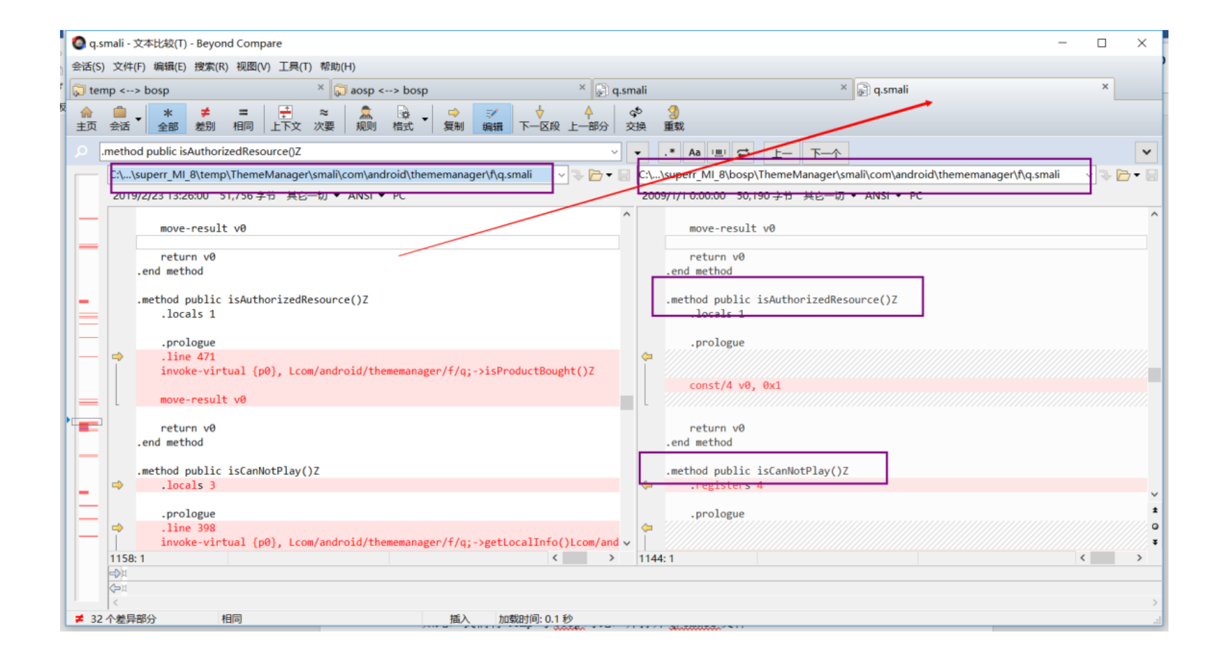
.method public isCanNotPlay()Z

.method public isProductBought()Z

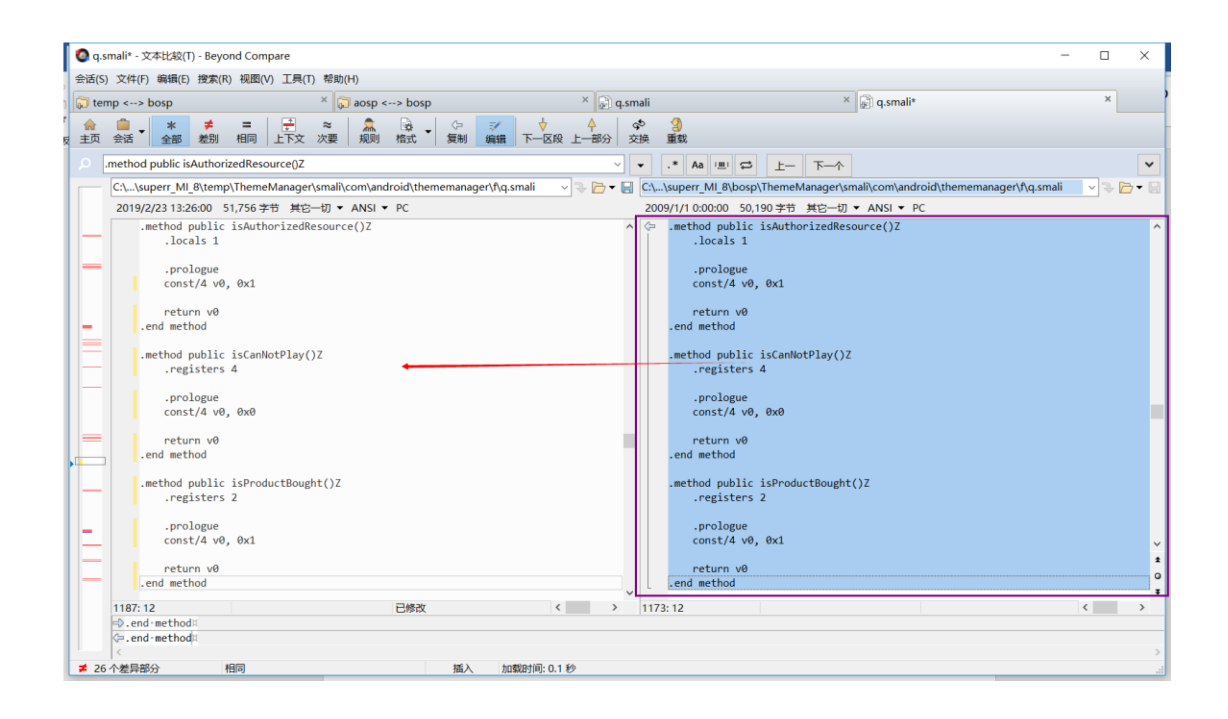
3 个方法直接返回为真：



如此，我们将 temp 与 bosp 对比，并打开 q.smali 文件



改为：



其余红色的不管（我 aosp 与 boap 中并未改动除这 3 处外的地方，我们这里同样的不做改动）

通过以上两个实例，总结下：辅助会自动启动 aosp 与 boap 和 temp 与 boap 对比窗口，前者对比，我们定位了需要修改的具体文件里面的具体方法，并得到了搜索关键字；后者对比，我们搜索关键字进行具体方法的修改。注意：我们要改的只有 temp（官方到反编译后的文件）内的相关文件。本质就是个插桩过程中，手动解冲突的过程，具体可访问，其原理是一致的：冲突解决-对、定、调：

https://v.youku.com/v_show/id_XODI1OTE1NzA0.html?spm=a2h0j.11185381.listitem_page1.5!2~A

编译辅助：https://share.weiyun.com/5KHB26O
这是什么？
简易来说，可称为通用的小白式作包辅助工具（支持 a/b 分区）
通用辅助（使用于航母，以及其他各类工具解包后使用）
厨房辅助（仅用于 win 下的捐赠版厨房）

能干什么？
主要有以下 3 个重点功能:
自动精简系统
自动反编译与回编译并复制回去原路径
自动启用对此器修改 smali 文件

那怎么用呢？
厨房辅助:解压本工具，复制到厨房目录，注意保持 Run.bat 与 superr.exe 同路径
通用辅助:解压本工具，复制到你的工具目录，注意保持 Run.bat 与 system 文件夹同路径

九、我没捐赠厨房，教程是不是没得用，没得玩了

这个，看你怎么想了，活学活用很重要。没捐赠厨房，我们有通用的辅助工具，不依赖厨房。至于解包工具，也不止厨房一家，此外还有免费版厨房，没有免费版厨房的辅助？把免费版解包合并后的 system 文件夹复制出来，用通用辅助就 ok

十、免费版厨房简易配置

链接与教程：https://share.weiyun.com/5FEvEUS
这里注意下，免费版厨房与捐赠版厨房界面几乎是一致的，具体操作可参考以上捐赠版厨房的使用教程

小宛 2019.3.5